described hereinabove to determine quickly whether the graphical data associated with a received command is to be rendered to the display device 83a-83d that is coupled to the unit 95a-95d. If the graphical data of the received command is not to be rendered to the display device 83a-83d coupled to the graphical acceleration unit 95a-95d, then the master server 55 of the graphical acceleration unit 95a-95d may be configured to discard the command before transmitting the graphical data of the command to any of the slave pipelines 56-59 and/or before performing any significant processing of the command. However, if any of the graphical data of the received command is to be rendered to the display device 83a-83d coupled to the graphical acceleration unit 95a-95d, then the unit 95a-95d can be configured to further process the command as described herein.

It should be noted that the system 350 could be scaled as needed in order to achieve a desired level of processing speed and/or image quality. In this regard, the number of graphical acceleration units 95a-95d and associated display devices 83a-83d can be increased or decreased as desired depending on how large or small of a single logical screen is desired. Further, the number of slave pipelines 56-59 (FIG. 3) within each graphical acceleration unit 95a-95d can be increased or decreased based on how much processing speed and/or image quality is desired for each display device 83a-83d. Note that the number of slave pipelines 56-59 within each unit 95a-95d does not have to be the same, and the modes and/or the combinations of modes implemented by each unit 95a-95d may be different.

Furthermore, in the embodiment shown by FIG. 3, mode inputs from the user were provided to the master pipeline 55, which controlled the mode of operation of the slave pipelines 55-59 and the compositor 76. In the embodiment shown by FIG. 14, such inputs may be similarly provided to the master pipeline 55 within each graphical acceleration unit 95a-95d via the client 52 and the SLS server 356. However, as previously set forth

hereinabove, there are various other methodologies that may be employed to control the mode of operation of the pipelines 56-59 and the compositor 76.

Preferred Embodiment

Having described an illustrative environment of a system embodying the present invention, reference is not made to the preferred embodiment of the present invention. In this regard, is should be understood that the foregoing discussion should not be viewed as limiting upon the invention, but rather as illustrative of only one system in which the present invention may reside and operate.

Having described a particular embodiment of a multiple-processor, single display system that may utilize the present invention, reference will now be made to various embodiments of the present invention itself. In this regard, reference is made to FIG. 16, which is a diagram illustrating certain principal components of the system 300 constructed in accordance with one embodiment of the invention. As summarized above, the present invention relates to systems and methods for configuring multiple computers to cooperatively operate to process and render a single display. The embodiment of FIG. 16 illustrates a two-tiered system having a master computer 302 and a plurality of slave computers 304, 306, 308, and 310 that may inter-communicate across a network.

In accordance with one aspect of the invention, the master computer 302 is responsible for configuring each of the slave computers 304, 306, 308, and 310 such that they operate cooperatively to render a single display (not shown). It should be appreciated that the configuration of each slave computer 304, 306, 308, and 310 need not be identical, but rather compatible. In this regard, and as previously discussed, there are certain modes and graphics configurations (*e.g.*, "stereo" mode versus "mono" mode) whereby the various slave computers may be incompatibly configured. The configuration

system and methodology of the present invention ensures compatible operation among the plurality of slave computers. Further, it should be appreciated that the graphics cards that are present in each of the slave computers need not be identical.

In essence, the master computer 302 receives instructions regarding the configuration for the graphics display, translates that configuration information into a format that is appropriate for each of the individual slave computers, and then communicates that individualized configuration information to each of the slave computers. By way of example, slave computer 304 may have a different graphics card than slave computer 306. With knowledge of these differences, the master computer 302 may specify the configuration information for each of the slave computers 304 and 306 in a slightly different fashion. Implementation details such as these will be appreciated by persons skilled in the art and are not deemed to be limiting upon the present invention. Accordingly, such implementation details need not be described herein.

In accordance with one embodiment of the invention, configuration information may be stored in a master configuration file 320. Preferably, such a master configuration file 320 will be stored in a predetermined location and using a predetermined file name, such that the master computer 302 can readily retrieve this information. The master computer 302 may then operate to translate the configuration information stored in this master configuration file 320 into distinct configuration information that is communicated separately to each of the slave computers 304, 306, 308, and 310. In this regard, the master computer 302 may include a program segment or process 322 that operates to perform such a configuration translation. This process 322 may then be configured to operate to output, for example, separate configuration files 324 and 326 for the separate slave computers 304 and 306, respectively. In such an embodiment, the slave configuration files 324 and 326 may be stored in a predetermined or known location in